5-2008

DNA Comparison Chip: Circuit Design and Evaluation

Tushar Bajaj
University of Arkansas, Fayetteville

DNA Comparison Chip:

Circuit Design and Evaluation

Designer & Author: Tushar Bajaj, B.S.E.E

705 W. Putman Street, Apt# H4

Fayetteville, AR 72701

tbajaj@uark.edu

Advisor & Instructor: Dr. Randy L. Brown

Mentor: Dr. Alan H. Mantooth

3217 BELL Engineering Center

College of Engineering

University of Arkansas

Fayetteville, AR 72701

mantooth@uark.edu

Due Date: May 02, 2008

Table of Contents

I. List of Figures & Tables....……....3

II. Abstract……...4

III. Introduction……....5

IV. Background……....7

Biological Background...……..7

Electrical Background……...9

V. Microchip Design……..10

Working……..10

Designing……13

VI. Results and Simulation……..22

VII. Conclusion…….23

VIII. References…….24

IX. Acknowledgements……...24

X. Appendix……...25

List of Figures

Figure 1: DNA matching methodology……...6

Figure 2: Effect of 'X' in DNA matching methodology……..7

Figure 3: Structure of DNA double Helix……8

Figure 4: Working Diagram……...11

Figure 5: Block Diagram……...14

Figure 6: D-Flip Flop Schematic……...16

Figure 7: Shift Register……..17

Figure 8: DNA chip complete schematic……...18

Figure 9: D Flipflop layouts……...20

Figure 10: Layouts of Shift Registers……...…….20

Figure 11: DNA Comparison Microchip layout……....21

Figure 12: DNA chip Simulation……...…23

List of Tables

Table 1: Bit assignment for the Bases……10

Table 2: Primitive Cells and number of transistors……...13

Table 3: Dimensions of each Block and Cell…….15

Table 4: Output per Clock Cycle……...22

Abstract

DNA Comparison Chip: Design and Evaluation

By

Tushar Bajaj

The purpose of this project was to: First, design a microchip that is capable of comparing strings of DNA sequences. Second, evaluate the functioning of such a microchip and how it can implemented in various real world applications. Physically, this microchip will act as an attachment to any computer. The computer will send two strings of DNA sequences—one library string and one pattern string—and the chip will output the number of matches that exist between the two strings (at a given position of the pattern string with respect to the library string). The circuit design is based on a 600 nanometer process. Mentor Graphic Design tools were used for making the layouts, schematics and netlist comparisons. The circuit was successfully designed and the schematics were used to simulate the functionality of such a circuit. The simulations confirmed the working of the circuit as expected, in theory.

This chip can be used in several practical applications and does not require any extensive hardware to install it. Its applications extend in areas like: forensic science, genetic engineering, bioinformatics etc. The fundamentality of its purpose makes it a very useful and flexible device.

Introduction

This project accomplished two tasks: First, design and build a circuit of a microchip that will compare two strings of DNA and second, evaluate the application of such a device in various realms. Often DNA is compared to a blueprint of a living organism; it is like an encrypted map that has instructions on construction of other components of cells, such as proteins and RNA. Since proteins and things proteins make are responsible for everything that a living organism is made of, we can call DNA the building block of life. Different sequence of DNAs produce different proteins—this makes the decrypting of these DNA sequence imperative in order to understand how living organisms are formed and how they function.

This chip acts as a fundamental tool that helps us match different DNA sequences. These matching results help us to gain a deeper understanding of what sequence of DNA is responsible for what trait in an organism. "Pick any two people in the world, and you would find their DNA is 99.9 percent identical. The remaining 0.1 percent is the genetic basis of all of humanity's differences, from the shape of our faces to the way some people get cancer to the fact that some patients respond to a certain drug while others don't."[1] Structurally, DNA is a long polymer made of repeating units called nucleotides. Every nucleotide has a basic component; these basic components are of four types: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). It is the sequence in which these bases are placed in a DNA string that determines the functioning of that DNA (further elaborated in the next section).

Now, suppose we have two strings DNA sequences: one is a library sequence that is formed from an extensive database of DNA sequences, for example the Human Genome Project which has the DNA sequence for a human being (all human genes), which is approx. 3 billion bases. Let's name the second DNA sequence as the pattern sequence that we will compare with the

whole library and see how many matches exist between the two at various positions. Remember, in order to compare DNA sequences we need to compare the 4 bases. This is where the DNA comparison chip comes into play. Here is how the chip would conduct matching of the two strings:

Pattern String ⟶ Pos Q:G G T A T T G A A, Pos Y:G G T A T T G A A

Library String ⟶ A G C C T A G C C A T G A G A C G G G T A T C G A A T C G G A A C C T A G A

Figure 1: DNA matching methodology

The chip counts the number of matches that exist at each position; the number of matches is added through a tree of adders and the result is given out the output pins in the form of a binary number. Then the library string is shifted forward with each clock cycle and new output is obtained at the end of the clock cycle. For instance, in the figure given above:

When the pattern string is in Position Q with respect to the library string, there are 4 matches out of total 9 possible. And when the string reaches Position Y, the number of matches is 8 out of 9.

Suppose one clock cycle takes about 20 nanoseconds. Total time it would take to count the number of matches at each location in the DNA library of 3 billion bases is:

$$3 \text{ X } 10^9 \text{ X } 20 \text{ nsec} = 1 \text{ minute}$$

This makes the whole process of comparing DNA strings, very fast!

Also, sometimes in a DNA, it doesn't matter which nucleotide follows a certain specific sequence—this is because they all produce the same amino acid. This microchip provides the capability of specifying that nucleotide as 'X', which represents that the nucleotide doesn't matter and it will not effect the matching count. The following figure shows how this works:

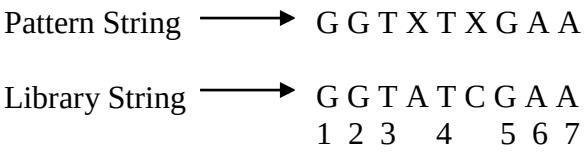


Figure 2: Effect of 'X' in DNA matching methodology

In the above configuration, since 'X' does not matter, there are 7 out of 7 possible matches.

All this functionality is implemented via single integrated circuit which is designed in a 600 nanometer process based on MOSIS design rules. Mentor^2 IC Design Manager was used to layout the whole circuit. The schematic of the circuit was made in Mentor Design Architect and it was simulated in Mentor QuickSim II. For testing, the simulation confirmed the working of the circuit as expected. Also in order to make sure that the schematic matched the layout, the netlist for the schematic was compared with the netlist of the layout through the LVS tool in Mentor IC Design Manager.

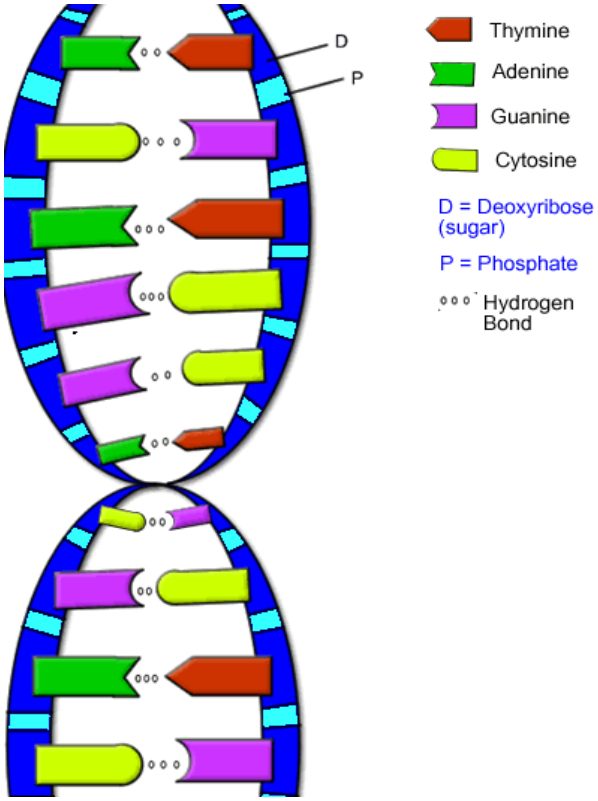
Background

Biological Background

Deoxyribonucleic acid or in short DNA is often called a blue print of a living organism. It contains genetic instructions that determine development and functioning of all living things known to man. DNA does this my mapping out the construction of other parts of a cell such as proteins and RNA(Ribonucleic acid).

The chemical structure of DNA is a long polymer made of basic units of nucleotides. The backbone of the polymer is made of sugars and phosphate groups, and the bonding is done through esters. Each sugar molecule is bonded to one of the four types of bases: Adenine (A),

Cytosine (C), guanine (G) and Thymine (T). It is the sequence of these bases that encodes the information.



In living organisms, DNA usually exists as a tightly associated pair of molecules. This forms a double helix, which is stabilized by the hydrogen bonds between the bases attached to the two strands.

Encrypted in DNA is a genetic code which is translated into proteins by living cells. Every triplet of nucleotides in a nucleic acid sequence specifies an amino acid. In total, there are 20 different amino acids used by living organisms to create proteins. These triplets are called codons. Figure 3: Structure of DNA double Helix If we look at the permutation and combination of

using 4 bases to form a codon of three, we come to the conclusion that there are $4^3 = 64$ codon combinations possible. In actuality, all 64 codons are responsible for either amino acids or start/stop signals during the process of formation. So the DNA controls not only what kind of amino acid, and hence protein, it will form but also how much will be made. We can correlate this phenomenon to a wireless signal: the signal sends the data in packets which includes start bit and stop bit; and the receiving end decodes this signal into sensible data. Also, like in a wireless signal the data carrier is a radio wave, for DNA translation in human body there is messenger RNA(mRNA). However, in RNA thymine(T) is

replaced by uracil(U). The following example clearly explains how this whole phenomenon works:

Lets take a RNA sequence: AUG UUU AGC UAG. This sequence begins with a start "command" amino acid (AUG-Methionine) and forms two protein forming amino acids—UUU(Phenylalanine) and AGC(Serine); and the final codon UAG(Amber) gives the stop "command".[3] The whole process is not so simple, many other factors(nearby sequences, initiation factors etc.) are involved, but this gives us a basic understanding of the matter that is relevant to this project.

The last aspect of DNA operations that is important to this project is "Degeneracy of genetic code". According to this phenomenon more than one sequence can be responsible for producing a same amino acid. For instance, GGA, GGC, GGG and GGU all form glycine. Here, it doesn't matter which nucleotide follows a sequence of GG in a triplet. The DNA comparison microchip designed in this project provides the capability of specifying that nucleotide as 'X'; which would mean that an 'X' has no effect in matching counts. Such a position in a DNA codon is called a degenerate site.

Electrical Background

The DNA comparison microchip is designed using MOSIS design rules. MOSIS(Metal Oxide Semiconductor Implementation Service) provides access to fabrication of prototype and low-volume production quantities of Integrated Circuits. If a chip to be manufactured by MOSIS , it should be build on design rules specified by them. The manufacturing process used is 600 nanometer; this number determines the wavelength of the electron beam used for the photolithography.

Mentor[3] Graphics Design tools were used to make the circuit. The tools from Mentor design suite that were used are:

- IC design manger for making the circuit layout
- Design architect for making the circuit schematic
- QuickSim II for running the simulations
- IC extract for extracting the netlist from the layout
- IC trace to compare the schematic and the layout netlists

Microchip Design

Working

Fundamentally, the objective of the design is to sum the number of matches that exist between the library DNA string and the pattern DNA string that is entered by the user, the total sum is to be produced at the output pins in a binary form. Since, the choices for comparing are only four we can assign each of them a two bit binary number:

	A	B
Adenine (A):	0	0
Guanine (G):	0	1
Thymine (T):	1	0
Cytosine (C):	1	1

Table 1: Bit assignment for the Bases

Hence, a two bit shift register is assigned for each library and pattern string, plus a one bit shift register is assigned for ‘X’ values. The following diagram clearly explains the functioning of the whole DNA comparison microchip:

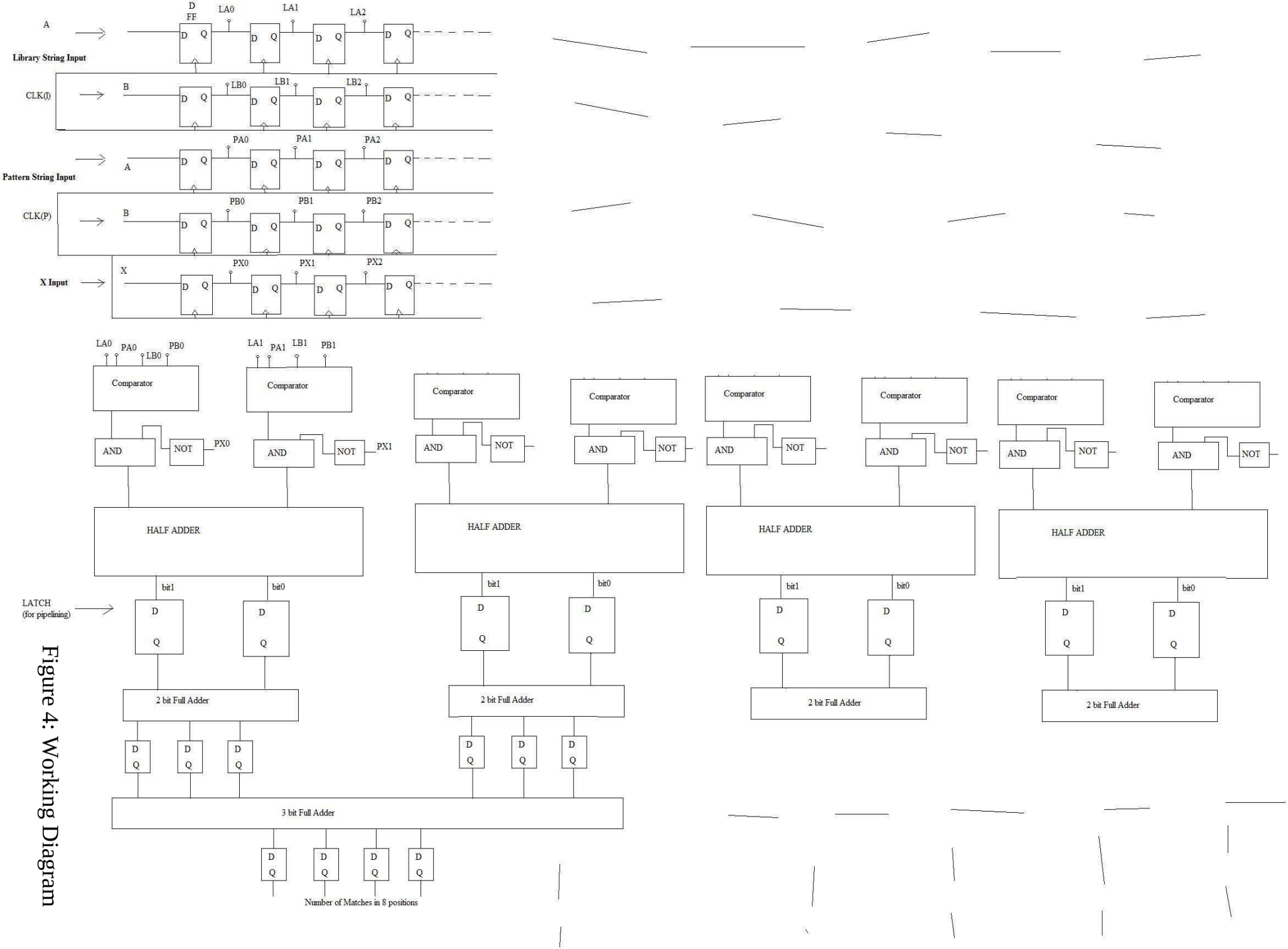


Figure 4: Working Diagram

The pattern and library shift registers have their independent clocks. This is because once the pattern shift register has been loaded it may not shift when the library shifts. Independent clocks give us the capability of loading and unloading the shift registers individually. Also, the clock for the 'X' shift register is also connected to the clock of pattern shift register. This is because 'X' appears in the pattern string only. Now, with figure 4 as our reference, let us discuss how the data flows in this circuit.

With every clock cycle, the library and pattern string (depending on which clock is changing) move forward in the 2 bit shift registers. The output of every D flipflop (DFF) in the shift register is fed into the comparators; these comparators do bit by bit comparison of each base from the respective strings at the base's respective position. If the bases match, the comparator outputs a '1' else a '0' for each base. This output is then ANDed with inverted value of X, i.e. if an 'X' exists (value=1) at a position, its inverted value is zero and when zero is ANDed with the output of the comparator, the output of the AND gate will always be '0', regardless of the output coming in from the comparator. The resulting output, like one more coming from the comparison of a base at the next position is fed into a half-adder. The result is then added(half adder) and fed into a latch(made of D flipflop). Then the result of D flipflop is fed into a 2bit full adder which adds the matches from 4 comparisons. Every output of an adder is fed into a latch, this pipelines the flow of data, i.e. the data shifts, and hence the output changes, only with the clock cycle.

Depending on the requirement, this system of adders and latches can be extended both horizontally and vertically to increase the number of matches it can conduct. In figure 4, the last output shown is coming from a 3 bit full adder; this output represents the number of matches that exist in the first 8 positions of both shift registers.

Designing

This is a design of a DNA comparison chip that is capable of conducting maximum 32 matches at any given instant. Since the chip is designed to be manufactured by MOSIS, the dimension of the final chip is 1.5mm X 1.5mm (a standard used by MOSIS for low cost fabrication by combining designs from many customers onto multi-project wafers). The process used is 600nm.

Step I: Floor Planning

This is a very basic and fundamental step that every design procedure should go through before beginning the actual layouts. Floor planning involves determining the size of each cell(primitive circuit) so that the whole circuit fits in the dimensions available to the designer. While doing the floor planning, some area should be reserved for the pad circuit. For this design, MOSIS AMI C5N pad circuit is used (see Appendix). This pad-ring leaves a square area of 900X900 μm^2 to build the layout of the whole circuit. Since process is 600nm or 0.6μm, wavelength (λ) = 0.3μm.

Therefore, total area available for design(in λ) = (900x900)/(0.3x0.3) $\lambda^2 = 9 \times 10^6 \lambda^2$

Now this area is divided among every cell depending on its requirement. To do this we need to find out how many total transistors would be used in the design. Figure 5 shows the major blocks that would be built. From this block diagram, we can calculate total number of transistors in the circuit. The following table shows the primitive cells used in the circuit and the number of transistor needed for them:

Cell	Transistors
D-Flipflop	22
Comparator	38
Half-Adder	18
Full Adder	28

Table 2: Primitive Cells and number of transistors

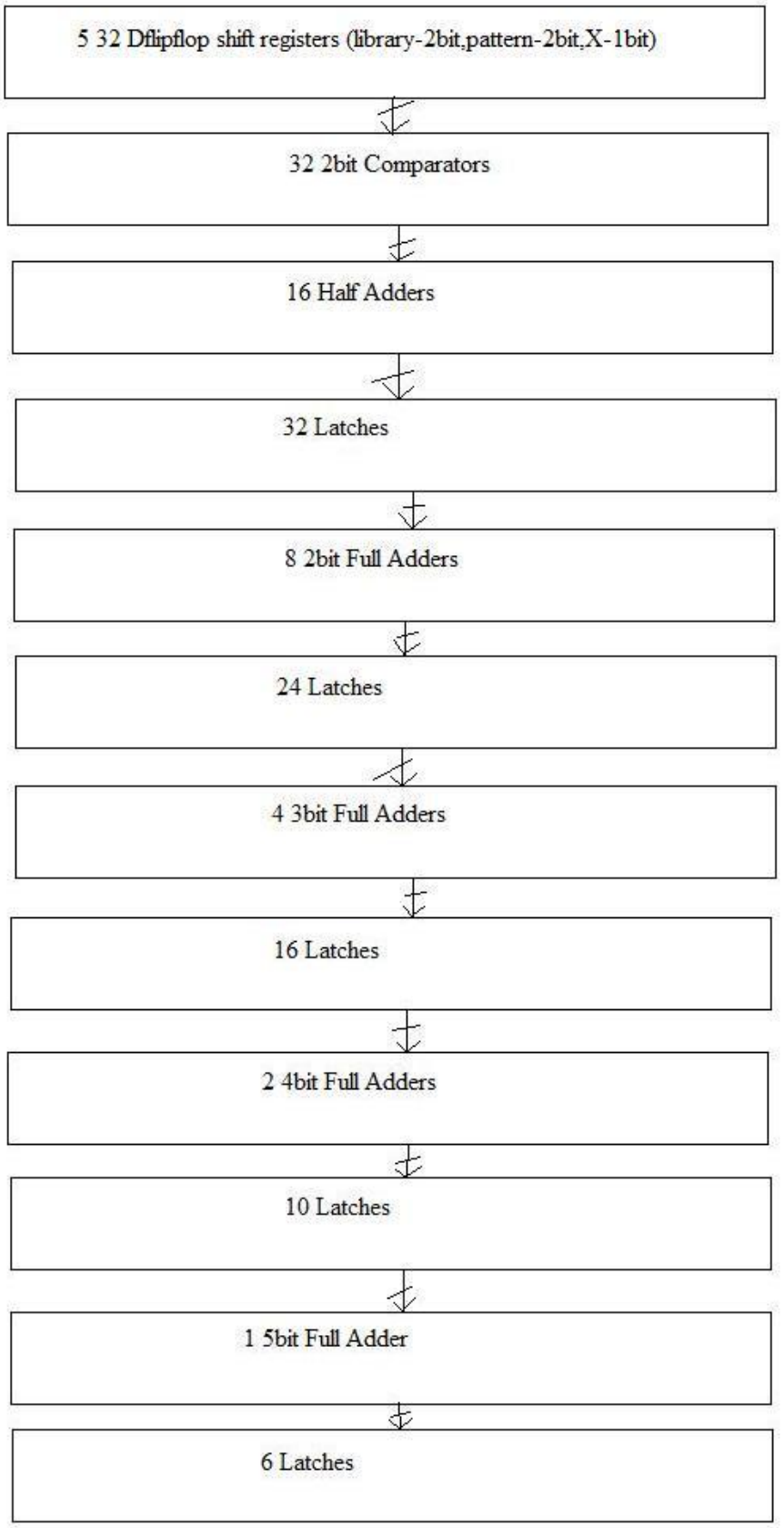


Figure 5: Block Diagram

Now this information is used to calculate the number of transistors in each block.

Shift register = 32 dflipflops = 32x22 trans. = 704 trans.

Shift register block = 5 shift registers = 5x704 = 3520 trans.

2 bit Comparator block = 32x38 trans. = 1216 trans.

Half Adders block = 16x18 trans. = 288 trans.

32 latches block = 32 dflipflops = 32x22 = 704 trans.

2bit Full Adder block = 8x2x28 = 448 trans.

24latches block=24x22=528 trans.

3bit Full Adder block = 4x3x28 = 336 trans.

16latches block=16x22=352 trans.

4bit Full Adder block = 2x4x28 = 224 trans.

10latches block=10x22=220 trans.

5bit Full Adder Block = 1x5x28 = 140 trans.

6latches block = 6x22 = 132 trans.

Therefore, Total number of transistors =

3520+1216+288+704+448+528+336+352+224+220+140+132 = 8108 transistors

Total area available = $9 \times 10^6 \lambda^2$

⇨ Area/transistor = $1110.0148 \lambda^2 = A$

⇨ Area of a Block =

(Number of transistor in the block) X (Area/transistor)

And max. Width of each block = 900 /0.3 = 3000 λ

⇨ Block Height = Block Area/3000

And Cell Width = Cell Area/ Block Height

The following table shows the total area available to each block and the primitive cell in that block; and height and width of the cell.

Block Name	**Number of Transistors**	**Total Block Area (λ^2)**	**Block/Cell Height (λ)**	**Area/Cell (λ^2)**	**Cell Width (λ)**
Shift Register	704	781450.42	260.48	24420.33	93.75
Comparator	1216	1349777.997	449.93	42180.56	93.75
Half Adder	288	319684.26	106.56	19980.27	187.50
32 Latches	704	781450.42	260.48	24420.33	93.75
2bit F.A	448	497286.63	165.76	62160.83	375.01
24 Latches	528	586087.81	195.36	24420.33	125.00
3bit F.A	336	372964.97	124.32	93241.24	750.01
16 Latches	352	390725.21	130.24	24420.33	187.50
4bit F.A	224	248643.32	82.88	124321.66	1500.00
10 Latches	220	244203.26	81.40	24420.33	300.00
5bit F.A	140	155402.07	51.80	155402.07	3000
6 Latches	132	146521.95	48.84	24420.33	500.00

Table 3: Dimensions of each Block and Cell

Note: The calculations and values given above do not take into account two clock buffers and extra space for wires. In the overall design, each cell has to be shortened so that there is an extra space to accommodate these features. The dimensions obtained during floor planning give an idea of what shape and size each cell can take.

Step II: Making the Schematic

Another step that is imperative in designing a chip is to first test the design by simulating it. In order to do this, a schematic of the whole circuit was made in Mentor Design Architect and then simulated using Mentor QuickSim II. The cells—Flipflops, Comparators,Gates and Adders were all built from the already available theoretical transistor models: PXFER(PMOS) and XFER(NMOS). These models are very basic and do not specify delays. They function in the following way:For XFER—if value at gate=1, then value at drain = source; for PXFER—if value at gate=0, then value of drain = source.

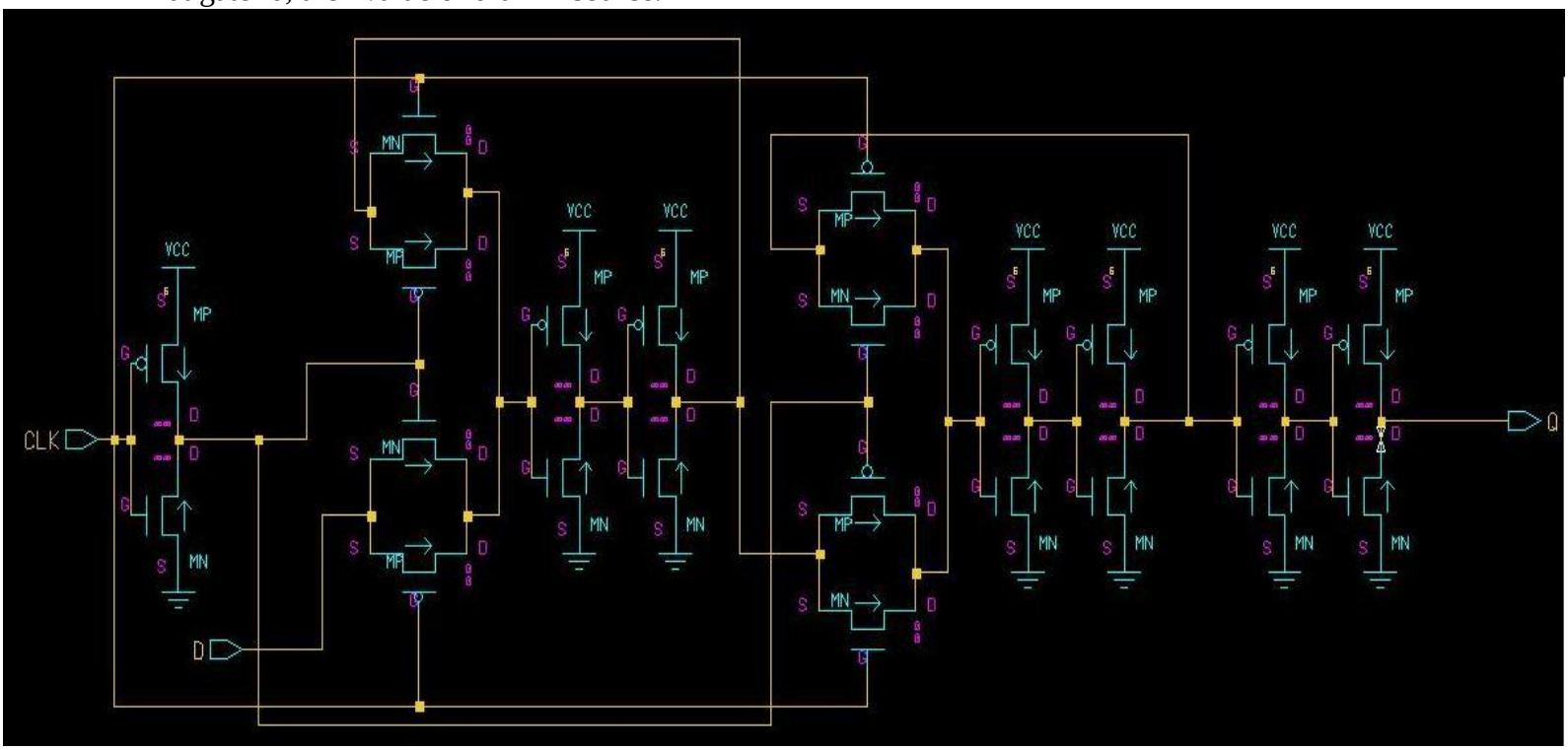


Figure 6: D-Flip Flop Schematic

The D flip flop schematic given in figure 6 shows how the primitive cells were built out of XFERs(labeled MN) and PXFERs(labeled MP).

The 32 bit shift registers are made by connecting 32 D flip flops in such a way that the output of one goes into the input of other. The following figure shows the schematic of a shift register:

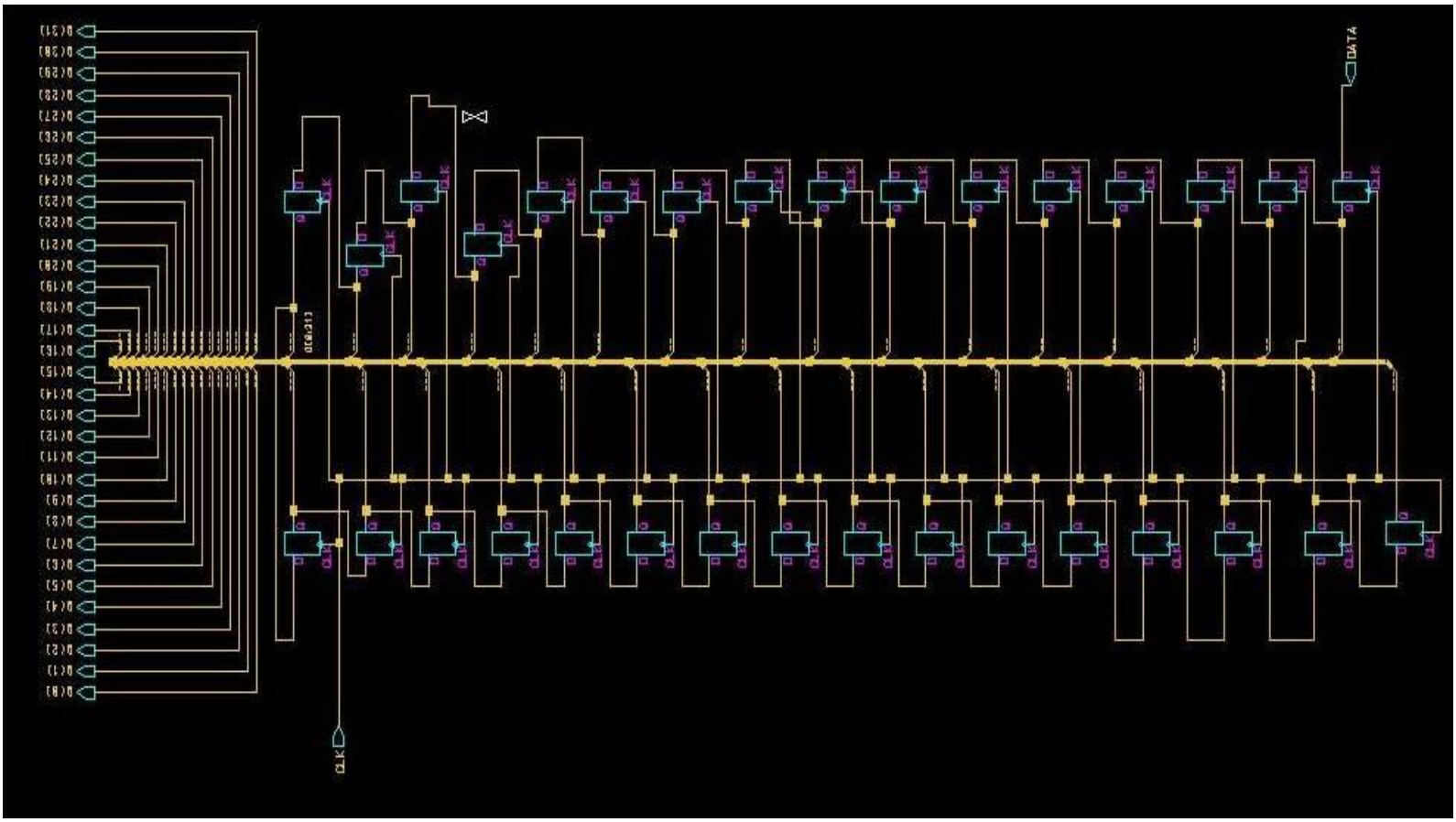


Figure 7: Shift Register

The schematics of other primitive cells are given in the Appendix.

These primitive cells are then assembled in a manner to produce the functionality described in the *Working* section. When combined with the pads, the schematic becomes complete:

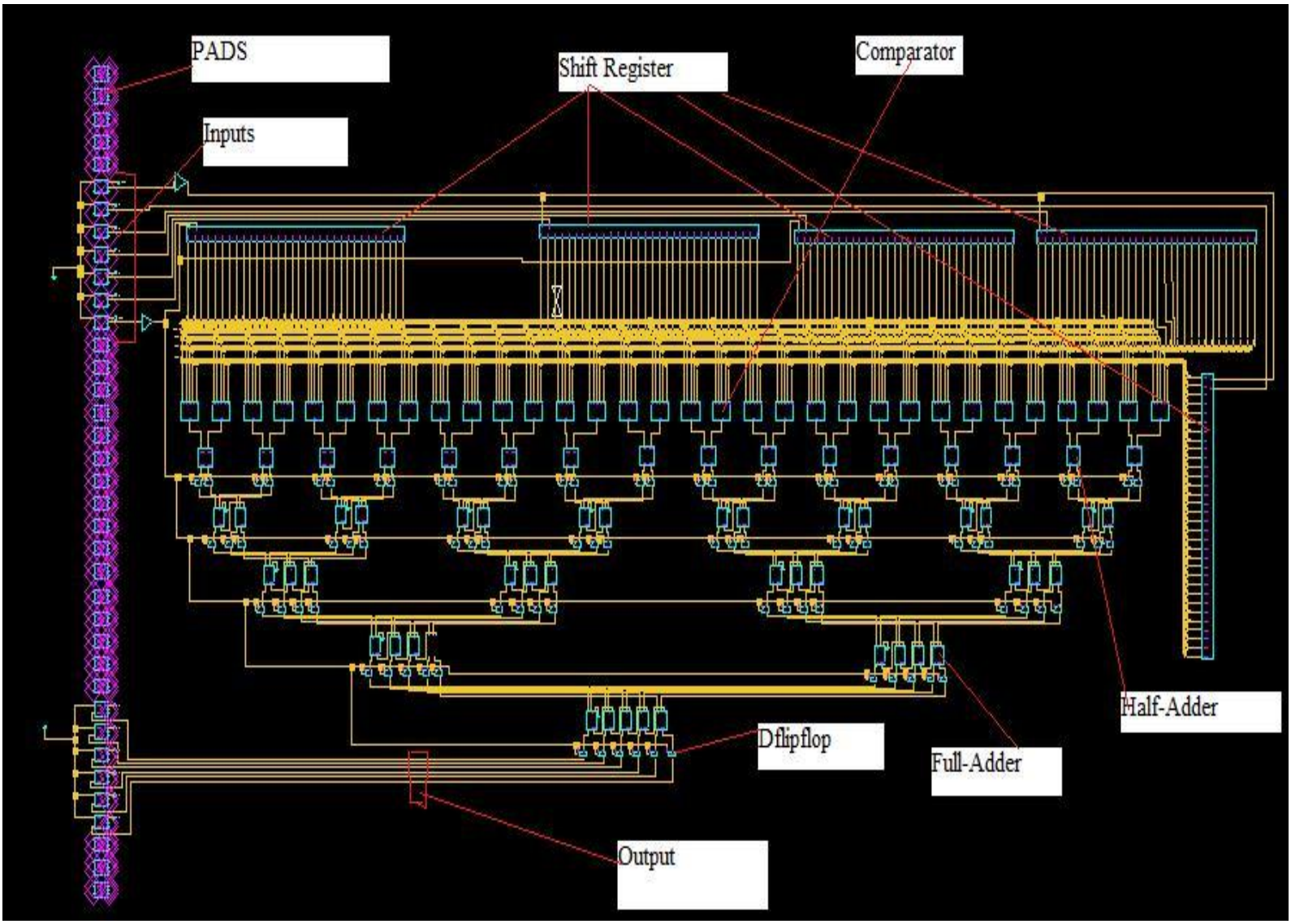


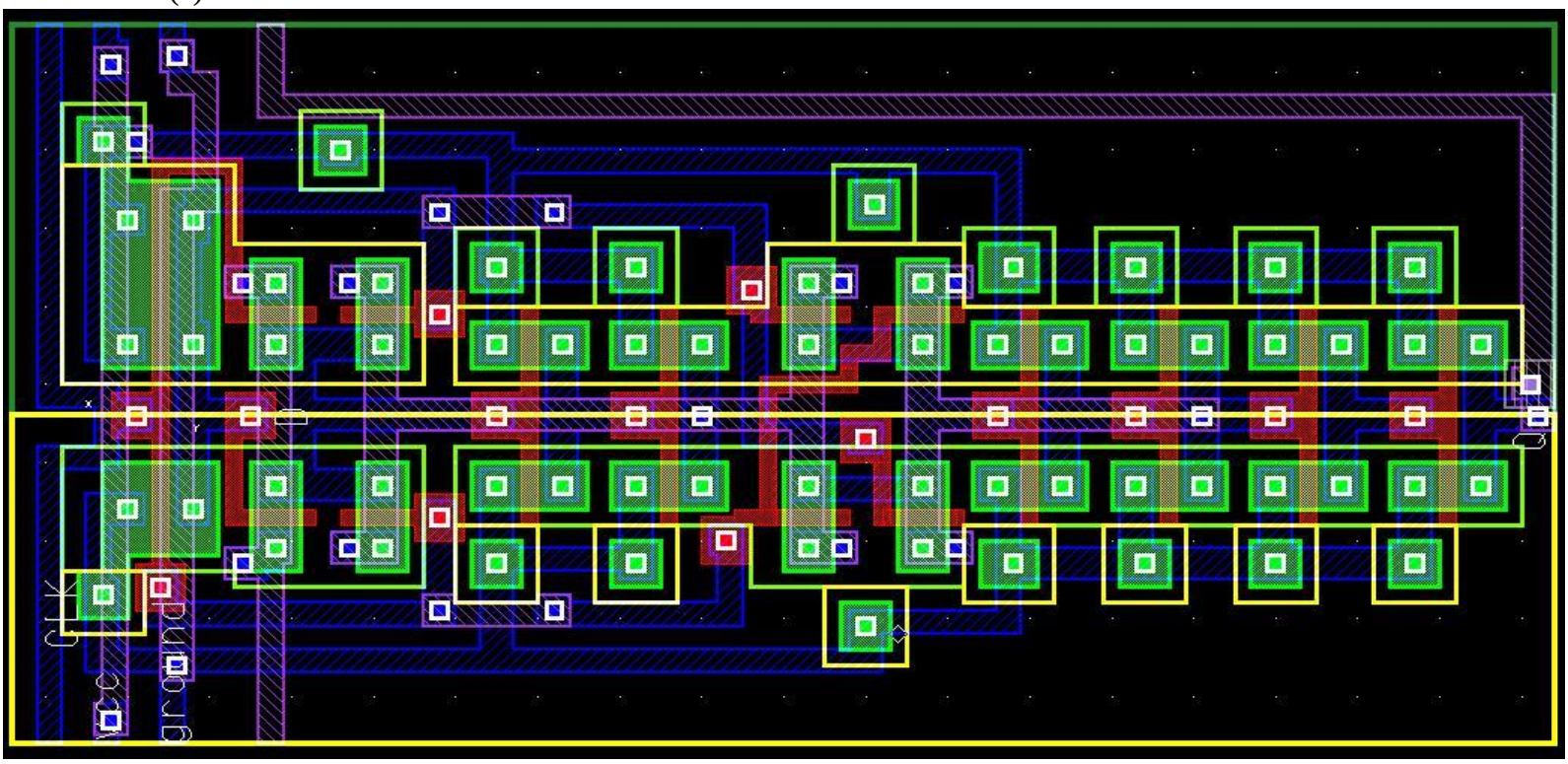
Figure 8: DNA chip complete schematic

Figure 8 shows the complete schematic of the DNA chip. This schematic is used to run simulations; and the results of these simulations are analyzed in order to predict if this system will work when under real world conditions.

Step III: Circuit Layout

The third and final step in the construction of the microchip was to layout the whole design. These steps were performed after the simulations from the schematic have confirmed that the chip will produce the desired results. The layouts are done using Mentor's IC design manager. Similarly to schematic, the layout was also built by first laying out the primitive cells. These cells were later joined together and re-arranged in a manner that they fit in the pad circuit layout provided by MOSIS (check appendix for the pad circuit layout). For the D flip flops and the full adders the height and width available per cell was different for different blocks of the layout, for instance, Height x Width for each cell of the shift register and 32 bit latches was 260.48x93.75; and for each cell of 6bit latches was 48.84x500. So, in the cases of latches and shift register, the primitive cells(D Flip Flop) were designed separately for each block according to the dimensions available. The following two figures show the layouts of the D Flip flops for the 32bit shift register and 6bit latches block:

(a)



(b)

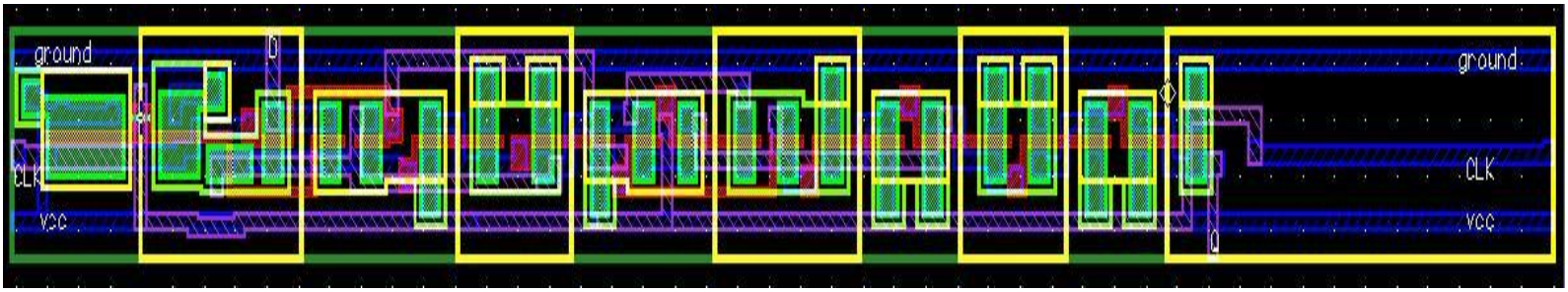
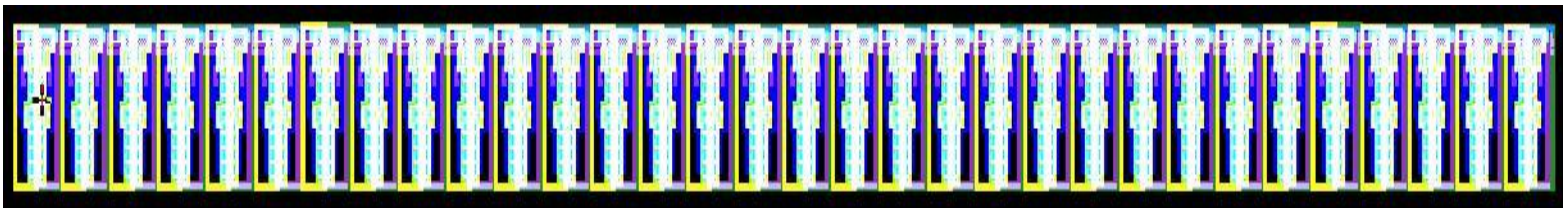


Figure 9: D Flipflop layouts for (a) 32bit shift register and (b) 6bit latch

Similarly, the D flip flops were designed separately for other blocks like: 32bit, 24bit, 16bit and 10bit latches. Same thing was done for the Full-Adder blocks, i.e. 2bit, 3bit, 4bit and 5bit Full-Adders. The D flip Flops were then used to form the shift registers and the latches:

(a)



(b)

Figure 10: Layout for (a) 32bit shift register and (b) 6bit latch

The layouts for the remaining primitive cells are given in the appendix. All the blocks are finally joined together and wires are connected from them to the pad pins to form the complete layout. The final DNA microchip layout is given on next page:

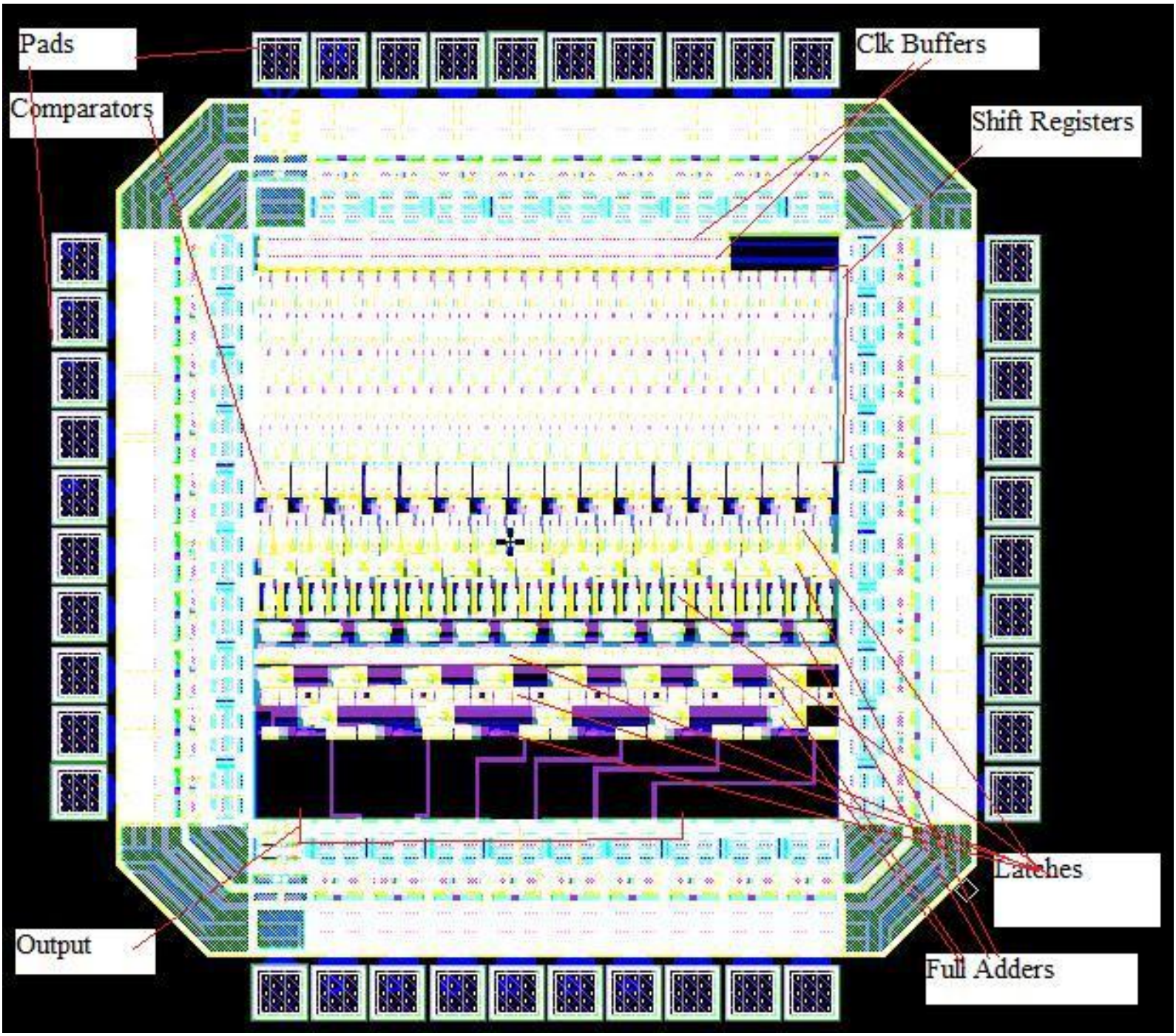


Figure 11: DNA Comparison Microchip layout

Results & Simulation

After the circuit was fully built, it was checked if it resembled the schematic, i.e. all the connections were made properly. This was done by comparing the netlists of the schematic and the layout. A built in tool called IC Extract acquires the netlist from the layout. And the LVS (layout versus schematic) tool of IC design manager was used to compare the two netlists. Both netlists matched successfully.

The schematic drawn in Mentor's Design Architect was run through simulations in Mentor's QuickSimII. The results obtained are as follows:

Following two strings were loaded in the pattern and the library shift registers-

Library: AGTCGTAGCTAGCTGCATAGCTAGCGATAGCGTAGCGTAG

Pattern: AGTCGATAGCXGATGCGTTGXATGXTGCAGTC

The DNA strings are entered from left to right. It takes 32 clock cycles to fully load the pattern string and 5 more clock cycle for the result to show at the output pins, i.e. the result of match at 32^{nd} clock cycle is viewed at 37^{th} clock cycle. Also, when the pattern shift register is completely loaded, its clock is stopped but the library clock is running until the whole library string is shifted through the library shift register. So, every clock cycle(37^{th} onwards) represent a specific output.

Clk Cycle	Pattern	Matches	Output
37	Lib: **AGTCG**TAGCTA**G**C**TGC**A**T**A**G**CTA**G**CGAT**AG**CGTAGCGTAG Pat: **AGTCG**ATAGCX**G**A**TGC**G**TT**G**XAT**G**XTGC**AG**TC	14	14
38	Lib: AGTCGT**A**GCTAGCTGCATAGCT**A**GCGATAGCGTAGCGTAG Pat: AGTCG**A**TAGCXGATGCGTTGX**A**TGXTGCAGTC	2	2
39	Lib: AGTCGTAGCTAGCTGCATAGCTAGC**G**A**T**AGC**GT**AGCGTAG Pat: AGTCGATAGCXGATGCGTTGXAT**G**X**T**GCA**GT**C	4	4
40	Lib: AGTC**GTAGCTAGC**T**G**CATA**G**C**T**AGCGATA**GC**GTAGCGTAG Pat: A**GTCGATAGC**X**G**ATGC**GTT**GXATGXT**GC**AGTC	12	12
41	Lib: AGTCGTAGCTAGCTGC**AT**AGC**T**A**G**CGATAGCGTAG**C**GTAG Pat: AGTCGATAGCXG**AT**GCG**T**T**G**XATGXTGCAGT**C**	5	5

Table 4: Output per Clock Cycle

The output appears on the pins Q[5..0]. Q5 being the most significant bit.

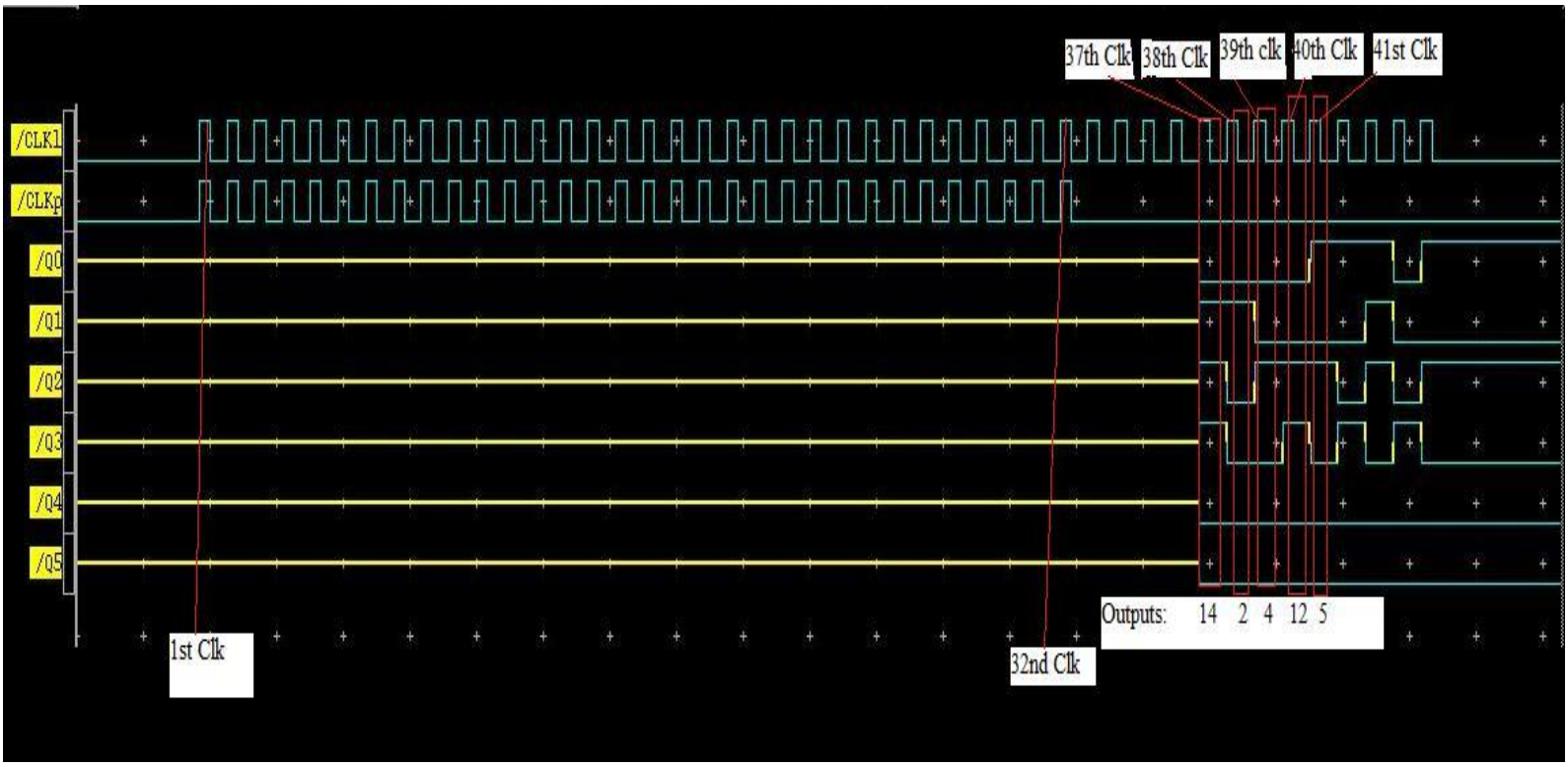


Figure 12: DNA chip Simulation

Conclusion

The results obtained from the simulations prove that this design should work properly when used in real world applications. Also, it is shown that the microchip can be made in a very compact from (1.5x1.5 mm^2).

This chip is ready to be sent for manufacturing through MOSIS.

Future Work: To test the manufactured chip in real world, calculate delays and check its sustainability.

References

[1] http://science.education.nih.gov/newsnapshots/TOC_Chips/Chips_RITN/chips_ritn.html

[2] Mentor Graphics: www.mentor.com

[3] http://en.wikipedia.org/wiki/Codon

http://ghr.nlm.nih.gov/handbook/basics/dna

CMOS VLSI Design: A Circuits and Systems Perspective, by Neil H.E. Weste & David Harris, Third Edition

Acknowledgements

This project could not have been possible without the support of following people:

Dr. Alan Mantooth: For his extraordinary support and guidance

Dr. Randy Brown: For Instilling the theoretical knowledge and providing guidance throughout the project.

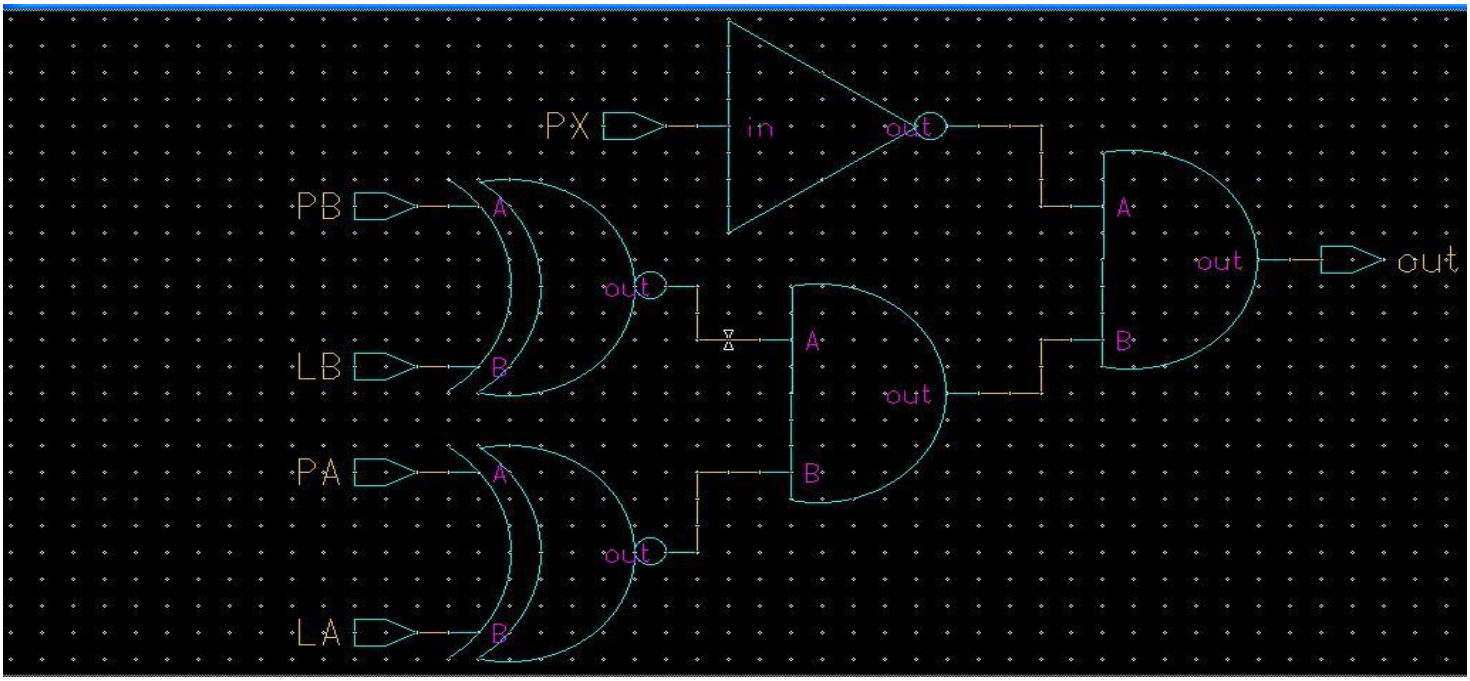
Kacie Thomas: For permitting to use her designs.

Michael Grubbs: For permitting me to use his designs.

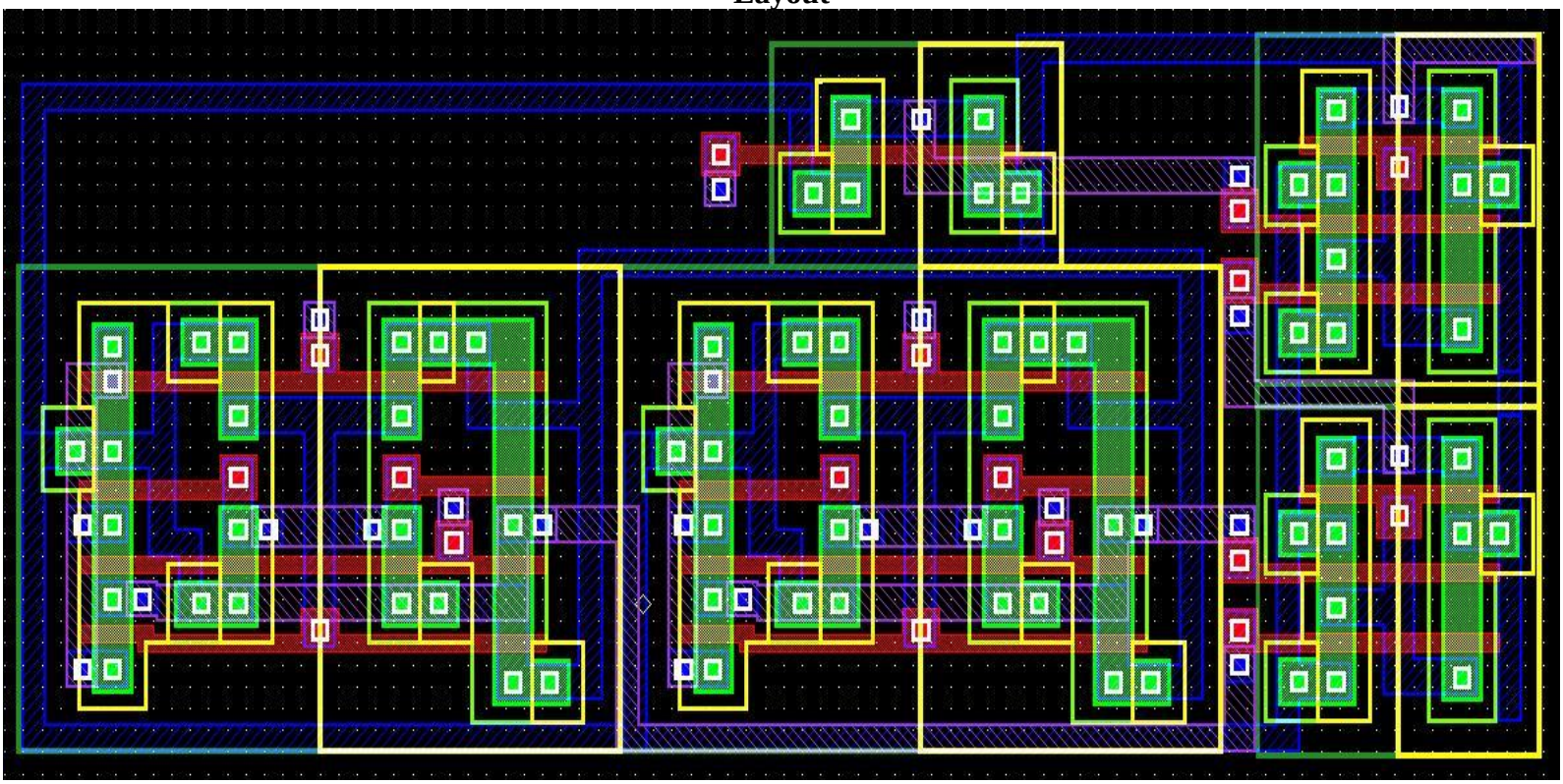
Appendix

(I) Comparator, Designed by Kacie Thomas

Schematic

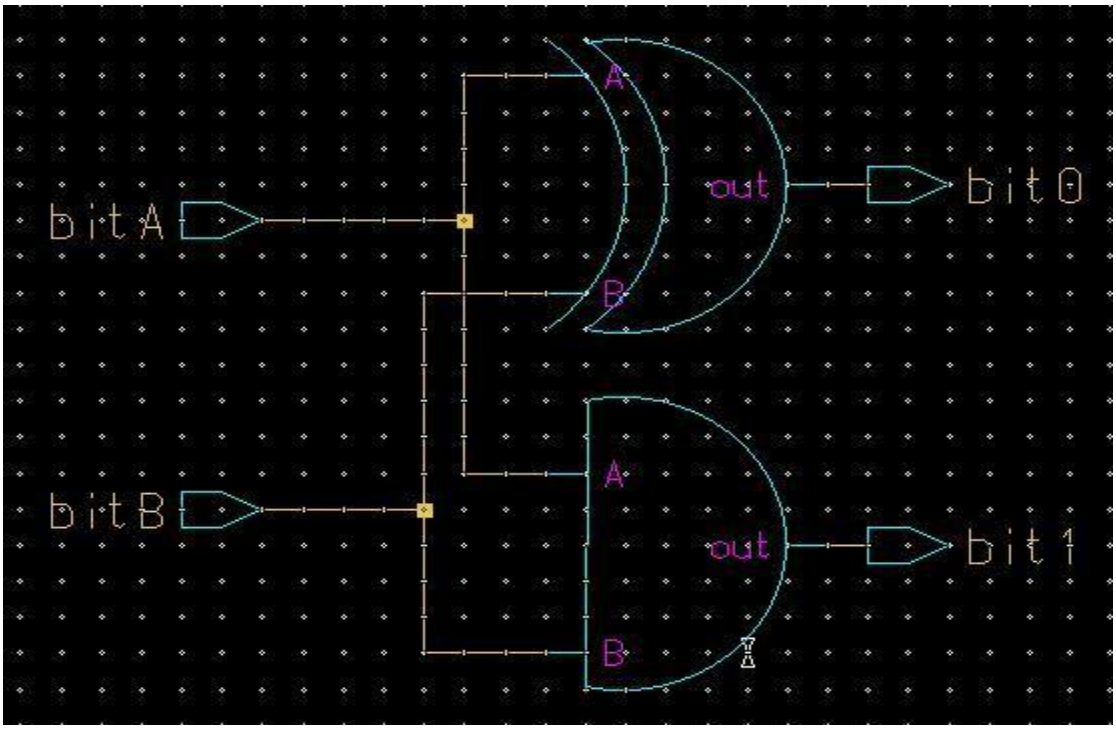


Layout

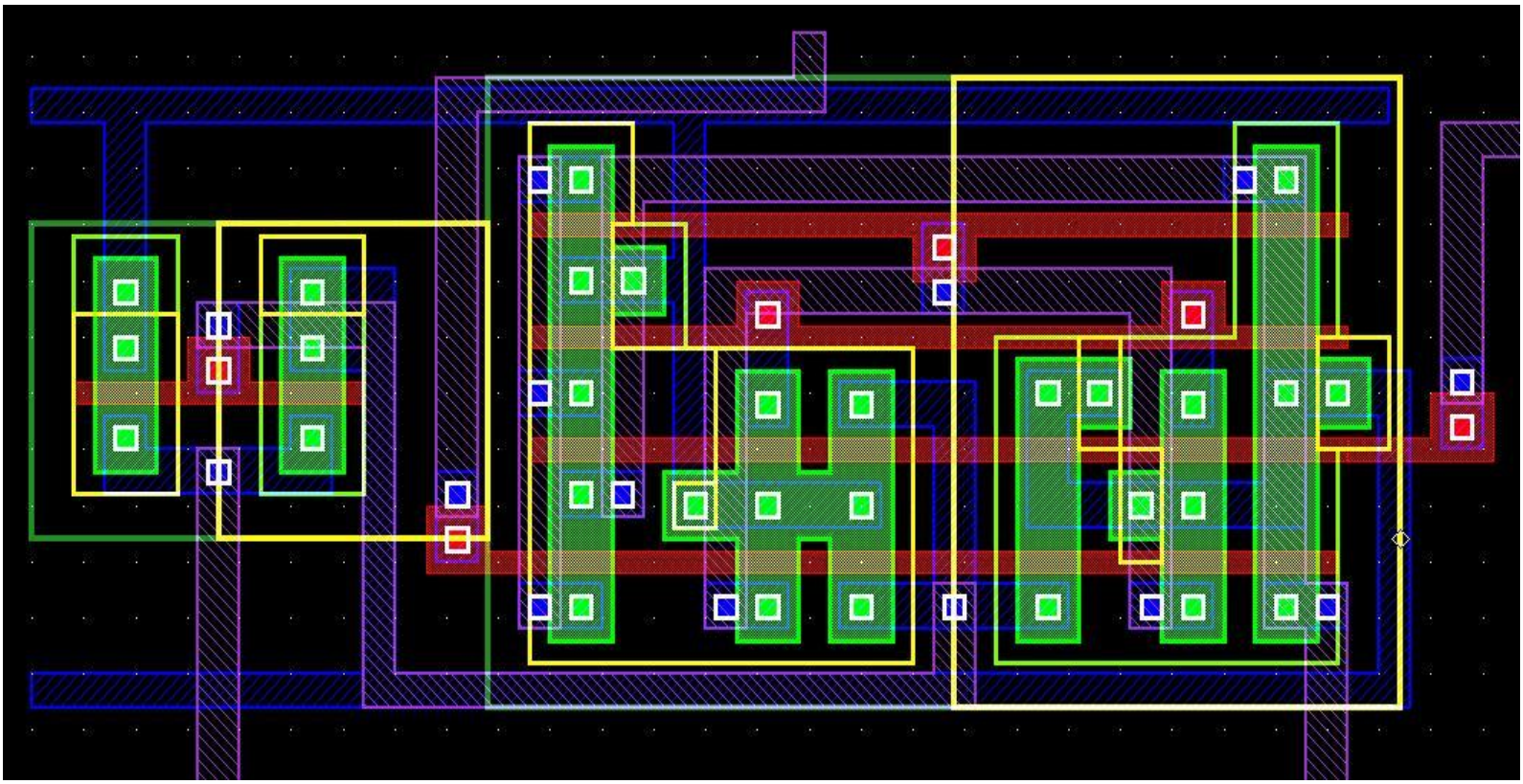


(II) Half-Adder, Designed by Kacie Thomas

Schematic

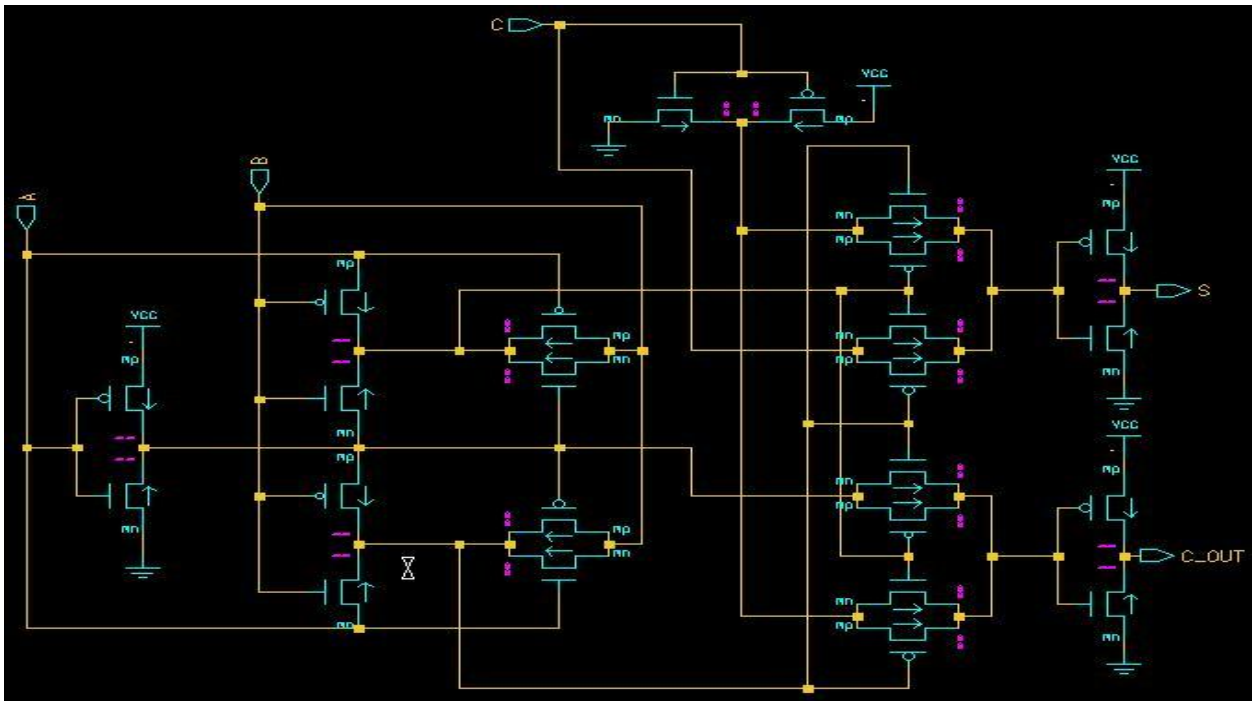


Layout

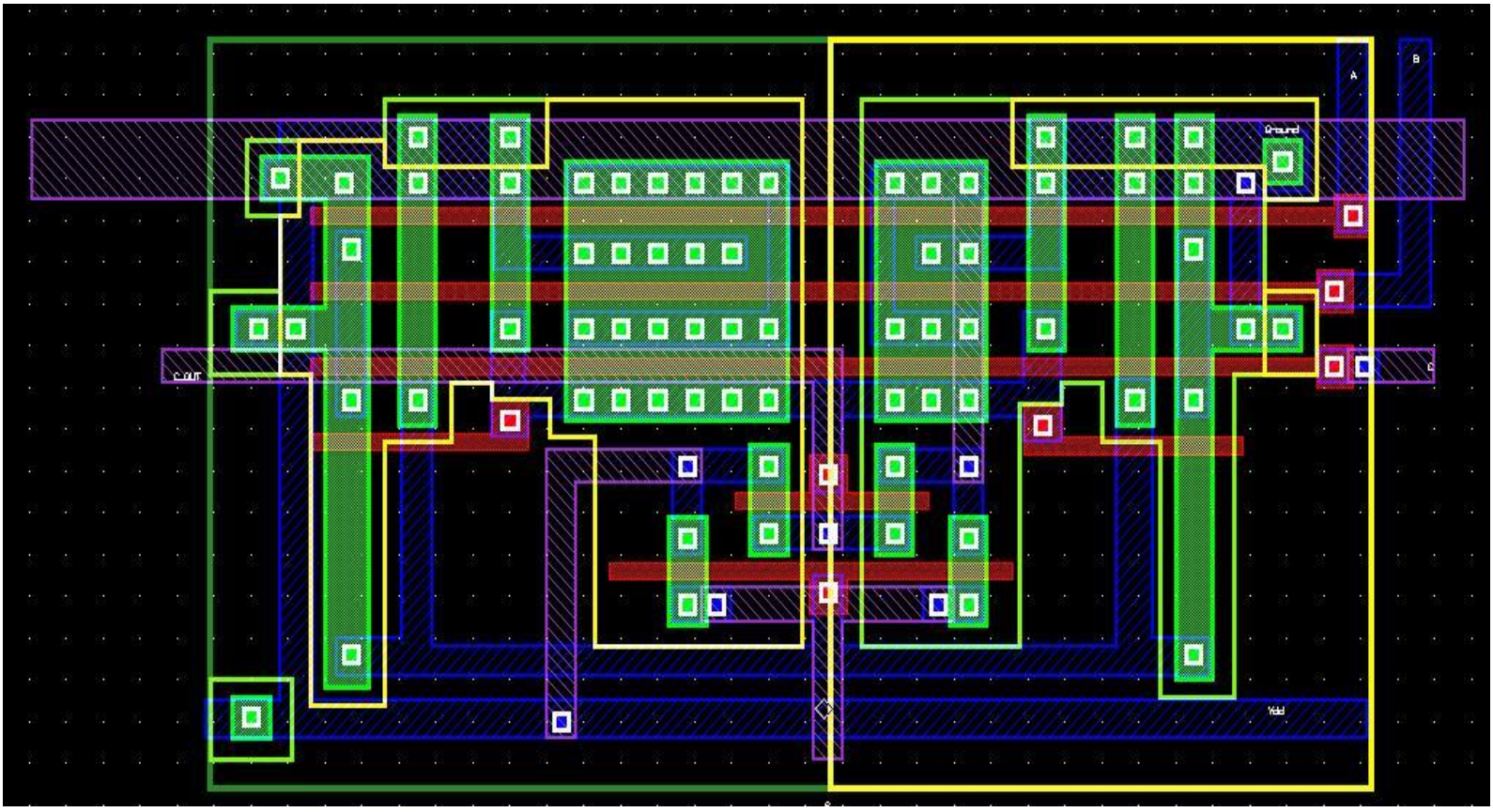


(III) Full-Adder, Designed by Michael Grubbs

Schematic



Layout



(IV) MOSIS AMI C5N pad circuit